

LIKELIHOOD CONSENSUS-BASED DISTRIBUTED PARTICLE FILTERING WITH DISTRIBUTED PROPOSAL DENSITY ADAPTATION

Ondrej Hlinka¹, Franz Hlawatsch¹, and Petar M. Djurić²

¹Institute of Telecommunications, Vienna University of Technology, Austria (ohlinka@nt.tuwien.ac.at)

²Department of Electrical and Computer Engineering, Stony Brook University, NY, USA (djuric@ece.sunysb.edu)

ABSTRACT

We present a consensus-based distributed particle filter (PF) for wireless sensor networks. Each sensor runs a local PF to compute a global state estimate that takes into account the measurements of all sensors. The local PFs use the joint (all-sensors) likelihood function, which is calculated in a distributed way by a novel generalization of the likelihood consensus scheme. A performance improvement (or a reduction of the required number of particles) is achieved by a novel distributed, consensus-based method for adapting the proposal densities of the local PFs. The performance of the proposed distributed PF is demonstrated for a target tracking problem.

Index Terms— Distributed particle filter, likelihood consensus, distributed proposal density adaptation, target tracking, wireless sensor network.

1. INTRODUCTION

We consider distributed sequential state estimation in a wireless sensor network. For general nonlinear/non-Gaussian scenarios, the particle filter (PF) is often the estimation method of choice [1]. In this paper, extending our work in [2–4], we propose a distributed PF that uses a novel distributed scheme for proposal density adaptation. As in [2–4], each sensor runs a *local* PF that computes a *global* state estimate incorporating the measurements of *all* sensors. The local PFs use the joint (all-sensors) likelihood function (JLF), which is computed in a decentralized way by means of the *likelihood consensus* (LC) scheme. Here, we present a generalized form of the LC originally proposed in [2], which is suited to a general measurement model (i.e., it is not limited to additive Gaussian measurement noises [2, 3] or likelihoods from the exponential family [4]).

Our main contribution is a novel distributed, consensus-based scheme for adapting the proposal densities (PDs) used by the local PFs. Adapted PDs can yield a significant performance improvement or, alternatively, a significant reduction of the required number of particles [5]. In our adaptation scheme, local PDs computed by the individual sensors are fused in a distributed way by means of consensus algorithms, thereby providing to each local PF a global PD reflecting all measurements. To make our scheme computationally feasible, we use Gaussian approximations for the local and global PDs. Our PD adaptation scheme differs from that proposed in [6] in that it is distributed and it uses Gaussian PD approximations.

Consensus-based distributed PFs with PD adaptation have been recently proposed in [7, 8]. The distributed PD adaptation scheme of [7] employs min- and max-consensus to construct a set capturing most of the posterior probability mass. This set is used to calculate a distorted state-transition density, which serves as PD. Our distributed PD adaptation scheme has a lower complexity than that of [7]. The communication requirements of our PD adaptation scheme

are somewhat higher, but the overall communication requirements of our distributed PF are still much lower than those of the distributed PF of [7], and simulation results demonstrate a better estimation performance of our distributed PF. In [8], a distributed unscented PF is presented. This method employs a PD adaptation which, however, is not distributed: the PD used at each sensor is only based on the local measurement. Again, simulation results demonstrate a better estimation performance of our distributed PF, which however comes at the cost of higher communication requirements.

This paper is organized as follows. In Section 2, we introduce the system model and review the principles of sequential Bayesian estimation. The LC-based distributed PF and the new generalized LC scheme are described in Section 3. In Section 4, we present the proposed distributed PD adaptation scheme. Finally, Section 5 reports simulation results for a target tracking problem.

2. SEQUENTIAL BAYESIAN STATE ESTIMATION

We consider a random, time-varying state vector $\mathbf{x}_n = (x_{n,1} \dots x_{n,M})^\top$. The state evolves according to the state-transition model

$$\mathbf{x}_n = \mathbf{g}_n(\mathbf{x}_{n-1}, \mathbf{u}_n), \quad n = 1, 2, \dots, \quad (1)$$

where \mathbf{u}_n is white driving noise with a known probability density function (pdf) $f(\mathbf{u}_n)$. At time n , \mathbf{x}_n is sensed by a sensor network with K sensors according to the measurement models

$$\mathbf{z}_{n,k} = \mathbf{h}_{n,k}(\mathbf{x}_n, \mathbf{v}_{n,k}), \quad k = 1, 2, \dots, K. \quad (2)$$

Here, $\mathbf{z}_{n,k}$ of dimension $N_{n,k}$ is the measurement at time n and at sensor k , and $\mathbf{v}_{n,k}$ is measurement noise with a known pdf $f(\mathbf{v}_{n,k})$. We assume that (i) $\mathbf{v}_{n,k}$ and $\mathbf{v}_{n',k'}$ are independent unless $(n, k) = (n', k')$; (ii) the initial state \mathbf{x}_0 and the sequences \mathbf{u}_n and $\mathbf{v}_{n,k}$ are all independent; and (iii) sensor k knows $\mathbf{g}_n(\cdot, \cdot)$ and $\mathbf{h}_{n,k}(\cdot, \cdot)$ for all n , but it does not know $\mathbf{h}_{n,k'}(\cdot, \cdot)$ for $k' \neq k$.

The state-transition and measurement models (1) and (2) together with our statistical assumptions determine the *state-transition pdf* $f(\mathbf{x}_n | \mathbf{x}_{n-1})$, the *local likelihood function* $f(\mathbf{z}_{n,k} | \mathbf{x}_n)$, and the JLF $f(\mathbf{z}_n | \mathbf{x}_n)$. Here, $\mathbf{z}_n \triangleq (\mathbf{z}_{n,1}^\top \dots \mathbf{z}_{n,K}^\top)^\top$ denotes the vector containing all sensor measurements at time n . Due to (2) and the independence of all $\mathbf{v}_{n,k}$, the JLF is given by

$$f(\mathbf{z}_n | \mathbf{x}_n) = \prod_{k=1}^K f(\mathbf{z}_{n,k} | \mathbf{x}_n). \quad (3)$$

Our goal is estimation of the state \mathbf{x}_n from all sensor measurements from time 1 to time n , $\mathbf{z}_{1:n} \triangleq (\mathbf{z}_1^\top \dots \mathbf{z}_n^\top)^\top$. To this end, we consider the minimum mean-square error (MMSE) estimator [9]

$$\hat{\mathbf{x}}_n^{\text{MMSE}} \triangleq \mathbb{E}\{\mathbf{x}_n | \mathbf{z}_{1:n}\} = \int \mathbf{x}_n f(\mathbf{x}_n | \mathbf{z}_{1:n}) d\mathbf{x}_n. \quad (4)$$

The posterior pdf $f(\mathbf{x}_n | \mathbf{z}_{1:n})$ in (4) can be calculated sequentially from the previous posterior $f(\mathbf{x}_{n-1} | \mathbf{z}_{1:n-1})$ and the JLF $f(\mathbf{z}_n | \mathbf{x}_n)$ [10]. A computationally feasible approximation to this *sequential*

This work was supported by the Austrian Science Fund (FWF) under Award S10603, by the NSF under Award CCF-1018323, and by the ONR under Award N00014-09-1-1154.

MMSE state estimation is provided by the PF, which represents the posterior pdf $f(\mathbf{x}_n | \mathbf{z}_{1:n})$ by a set of weighted particles [1].

3. LC-BASED DISTRIBUTED PARTICLE FILTER

The proposed LC-based distributed PF (LC-DPF) differs from our previous work in [2–4] by the generalized LC and the distributed PD adaptation (presented in Sections 3.2 and 4, respectively). As in [2–4], each sensor tracks a particle representation of the global posterior $f(\mathbf{x}_n | \mathbf{z}_{1:n})$ using a *local PF*. At each time n , each local PF obtains a state estimate $\hat{\mathbf{x}}_{n,k}$ that is based on $\mathbf{z}_{1:n}$, i.e., *all* sensor measurements up to time n . This requires knowledge of the JLF $f(\mathbf{z}_n | \mathbf{x}_n)$ as a function of \mathbf{x}_n . An approximation of the JLF is provided to all sensors in a distributed way by means of the generalized LC. No communication between distant sensors or complex routing protocols are required. Also, no particles, local state estimates, or measurements are communicated between the sensors. The proposed distributed PD adaptation scheme can yield a significant performance improvement or, alternatively, a significant reduction of the number of particles (and, thus, of the computational complexity); this comes at the cost of an increase in inter-sensor communications.

3.1. Local PF Algorithm

At a given time $n \geq 1$, the local PF at sensor k performs the following steps, which are identical for all k :

Step 1: A resampling [1] is applied to the J particles $\{\mathbf{x}_{n-1,k}^{(j)}\}_{j=1}^J$ with corresponding weights $\{w_{n-1,k}^{(j)}\}_{j=1}^J$ (calculated at time $n-1$) that represent the previous global posterior $f(\mathbf{x}_{n-1} | \mathbf{z}_{1:n-1})$ at sensor k . This produces J resampled particles $\{\bar{\mathbf{x}}_{n-1,k}^{(j)}\}_{j=1}^J$.

Step 2: Temporary particles $\{\mathbf{x}'_{n,k}^{(j)}\}_{j=1}^J$ are randomly drawn from $f(\mathbf{x}_n | \bar{\mathbf{x}}_{n-1,k}^{(j)}) \triangleq f(\mathbf{x}_n | \mathbf{x}_{n-1})|_{\mathbf{x}_{n-1} = \bar{\mathbf{x}}_{n-1,k}^{(j)}}$, and a Gaussian approximation $\mathcal{N}(\mathbf{x}_n; \boldsymbol{\mu}'_{n,k}, \mathbf{C}'_{n,k})$ of the “predicted posterior” $f(\mathbf{x}_n | \mathbf{z}_{1:n-1})$ is calculated according to

$$\boldsymbol{\mu}'_{n,k} = \frac{1}{J} \sum_{j=1}^J \mathbf{x}'_{n,k}^{(j)}, \quad \mathbf{C}'_{n,k} = \frac{1}{J} \sum_{j=1}^J \mathbf{x}'_{n,k}^{(j)} \mathbf{x}'_{n,k}^{(j)\top} - \boldsymbol{\mu}'_{n,k} \boldsymbol{\mu}'_{n,k}^\top. \quad (5)$$

Step 3 (jointly performed by all sensors, using communication with neighboring sensors): An adapted Gaussian PD $q(\mathbf{x}_n; \mathbf{z}_n) \triangleq \mathcal{N}(\mathbf{x}_n; \boldsymbol{\mu}_n, \mathbf{C}_n)$ involving all sensor measurements is computed from the $\boldsymbol{\mu}'_{n,k'}$, $\mathbf{C}'_{n,k'}$, and $\mathbf{z}_{n,k'}$ ($k' = 1, \dots, K$) by means of the distributed, consensus-based scheme described in Section 4.

Step 4: J particles $\{\mathbf{x}_{n,k}^{(j)}\}_{j=1}^J$ are drawn from the PD $q(\mathbf{x}_n; \mathbf{z}_n)$.

Step 5 (jointly performed by all sensors, using communication with neighboring sensors): An approximation $\tilde{f}(\mathbf{z}_n | \mathbf{x}_n)$ of the JLF $f(\mathbf{z}_n | \mathbf{x}_n)$ is computed in a distributed way by means of the generalized LC described in Section 3.2, using the particles $\{\mathbf{x}_{n,k}^{(j)}\}_{j=1}^J$ drawn in Step 4.

Step 6: Weights associated with the particles $\mathbf{x}_{n,k}^{(j)}$ drawn in Step 4 are calculated according to

$$w_{n,k}^{(j)} = \gamma \frac{\tilde{f}(\mathbf{z}_n | \mathbf{x}_{n,k}^{(j)}) f(\mathbf{x}_{n,k}^{(j)} | \mathbf{x}_{n-1,k}^{(j)})}{q(\mathbf{x}_{n,k}^{(j)}; \mathbf{z}_n)}, \quad j = 1, \dots, J,$$

where γ is chosen such that $\sum_{j=1}^J w_{n,k}^{(j)} = 1$.

Step 7: From $\{\mathbf{x}_{n,k}^{(j)}, w_{n,k}^{(j)}\}_{j=1}^J$, an approximation of the global MMSE state estimate $\hat{\mathbf{x}}_n^{\text{MMSE}}$ in (4) is computed according to

$$\hat{\mathbf{x}}_{n,k} = \sum_{j=1}^J w_{n,k}^{(j)} \mathbf{x}_{n,k}^{(j)}.$$

Initialization: The recursive procedure defined by Steps 1–7 is initialized at time $n=0$ by J particles $\mathbf{x}_{0,k}^{(j)}$ randomly drawn from an appropriate prior pdf $f(\mathbf{x}_0)$, and by equal weights $w_{0,k}^{(j)} \equiv 1/J$.

3.2. Generalized Likelihood Consensus

We now present the generalized LC scheme that is used in Step 5 to provide an approximate JLF to each sensor. In contrast to our previous work [2–4], this scheme is not limited to likelihoods with exponential form or to additive Gaussian measurement noises; it is suitable for any type of likelihood and any measurement model (2).

To derive the generalized LC, we first take the logarithm of (3):

$$\log f(\mathbf{z}_n | \mathbf{x}_n) = \sum_{k=1}^K \log f(\mathbf{z}_{n,k} | \mathbf{x}_n). \quad (6)$$

Unfortunately, a consensus-based distributed calculation of (6) is not possible in general because the terms of the sum depend on the unknown state \mathbf{x}_n . We therefore use the following approximate (finite-dimensional) basis expansions of the local log-likelihoods:

$$\log f(\mathbf{z}_{n,k} | \mathbf{x}_n) \approx \sum_{r=1}^R \alpha_{n,k,r}(\mathbf{z}_{n,k}) \varphi_{n,r}(\mathbf{x}_n). \quad (7)$$

Here, $\alpha_{n,k,r}(\mathbf{z}_{n,k})$ are expansion coefficients that contain all sensor-local information (including the sensor measurement $\mathbf{z}_{n,k}$) and $\varphi_{n,r}(\mathbf{x}_n)$ are fixed, sensor-independent basis functions that are assumed to be known to all sensors. Substituting (7) into (6), we obtain

$$\log f(\mathbf{z}_n | \mathbf{x}_n) \approx \sum_{r=1}^R a_{n,r}(\mathbf{z}_n) \varphi_{n,r}(\mathbf{x}_n), \quad (8)$$

with

$$a_{n,r}(\mathbf{z}_n) = \sum_{k=1}^K \alpha_{n,k,r}(\mathbf{z}_{n,k}). \quad (9)$$

The sum over all sensors in (9) can be easily computed in a distributed way by means of a consensus algorithm [11] since the terms of the sum are real numbers (not functions of \mathbf{x}_n).

By exponentiating (8), we finally obtain the following approximation of the JLF, denoted $\tilde{f}(\mathbf{z}_n | \mathbf{x}_n)$:

$$f(\mathbf{z}_n | \mathbf{x}_n) \approx \tilde{f}(\mathbf{z}_n | \mathbf{x}_n) = \exp \left(\sum_{r=1}^R a_{n,r}(\mathbf{z}_n) \varphi_{n,r}(\mathbf{x}_n) \right). \quad (10)$$

Therefore, a sensor that knows the coefficients $a_{n,r}(\mathbf{z}_n)$ is able to evaluate the approximate JLF $\tilde{f}(\mathbf{z}_n | \mathbf{x}_n)$ for all values of \mathbf{x}_n . In fact, the vector of all coefficients, $\mathbf{a}_n(\mathbf{z}_n) \triangleq (a_{n,1}(\mathbf{z}_n) \cdots a_{n,R}(\mathbf{z}_n))^\top$, can be viewed as a *sufficient statistic* [9] that epitomizes the total measurement \mathbf{z}_n within the limits of the approximation (7). The expressions (9) and (10) allow a distributed, consensus-based calculation of $\tilde{f}(\mathbf{z}_n | \mathbf{x}_n)$ due to the following key facts. (i) The coefficients $a_{n,r}(\mathbf{z}_n)$ do not depend on the state \mathbf{x}_n but contain the information of all sensors (i.e., the expansion coefficients $\alpha_{n,k,r}(\mathbf{z}_{n,k})$ for all k). (ii) The state \mathbf{x}_n enters into $\tilde{f}(\mathbf{z}_n | \mathbf{x}_n)$ only via the basis functions $\varphi_{n,r}(\cdot)$, which are sensor-independent and known to each sensor. (iii) According to (9), the $a_{n,r}(\mathbf{z}_n)$ are sums in which each term contains only local information of a single sensor.

At each time n , the expansion coefficients $\alpha_{r,n,k}(\mathbf{z}_{n,k})$ in (7) are calculated locally at each sensor k by means of *least squares fitting* [12] based on the J data points $\{\mathbf{x}_{n,k}^{(j)}, \log f(\mathbf{z}_{n,k} | \mathbf{x}_{n,k}^{(j)})\}_{j=1}^J$. Here, the use of the particles $\mathbf{x}_{n,k}^{(j)}$ drawn in Step 4 of the local PF algorithm ensures a good approximation in those regions of the state-

space where the approximate JLF $\tilde{f}(\mathbf{z}_n | \mathbf{x}_n)$ is evaluated in Step 6. (We assume that $f(\mathbf{z}_{n,k} | \mathbf{x}_{n,k}^{(j)}) \neq 0$.)

The steps of the *generalized LC scheme* performed at a given time n can now be summarized as follows:

Step 1: Sensor k calculates the expansion coefficients $\alpha_{r,n,k}(\mathbf{z}_{n,k})$ in (7) using least squares fitting.

Step 2: The coefficients $\alpha_{r,n,k}(\mathbf{z}_{n,k})$ of all sensors k are added in a distributed way using a consensus algorithm. One instance of that algorithm is employed for each $r \in \{1, \dots, R\}$; all instances are executed in parallel. After a sufficient number of consensus iterations, the $a_{r,n}(\mathbf{z}_n)$ (see (9)) for all r are available at each sensor.

Step 3: Using the $a_{r,n}(\mathbf{z}_n)$, each sensor is able to evaluate the approximate JLF $\tilde{f}(\mathbf{z}_n | \mathbf{x}_n)$ for any value of \mathbf{x}_n according to (10).

The proposed LC-DPF (without the PD adaptation described in Section 4) requires the transmission of IR real numbers by each sensor at each time n , where I is the number of consensus iterations performed by each consensus algorithm and R (cf. (7)) is the number of consensus algorithms executed in parallel. All transmissions are to neighboring sensors only, and their number does not depend on the measurement dimensions $N_{n,k}$. Thus, the LC-DPF is particularly attractive in the case of high-dimensional measurements.

4. DISTRIBUTED PROPOSAL ADAPTATION

We now present our distributed scheme for calculating the adapted PD $q(\mathbf{x}_n; \mathbf{z}_n)$ (Step 3 of the local PF algorithm in Section 3.1). This scheme can be summarized as follows. First, a “pre-distorted” local posterior is calculated at each sensor. The local posteriors are then fused via a distributed fusion rule to obtain a global posterior, which is used as PD by each local PF.¹ This PD takes into account the measurements of *all* sensors, which is appropriate in view of the fact that the JLF is used in Step 6. Our approach is inspired by the one from [13], which however was proposed in a different context and uses a fusion rule different from ours.

We first note that the global posterior $f(\mathbf{x}_n | \mathbf{z}_{1:n})$ can be written (up to a normalization factor) as

$$\begin{aligned} f(\mathbf{x}_n | \mathbf{z}_{1:n}) &= f(\mathbf{x}_n | \mathbf{z}_{1:n-1}, \mathbf{z}_n) \\ &\propto f(\mathbf{z}_n | \mathbf{x}_n, \mathbf{z}_{1:n-1}) f(\mathbf{x}_n | \mathbf{z}_{1:n-1}) \\ &= f(\mathbf{z}_n | \mathbf{x}_n) f(\mathbf{x}_n | \mathbf{z}_{1:n-1}) \\ &= \left[\prod_{k=1}^K f(\mathbf{z}_{n,k} | \mathbf{x}_n) \right] f(\mathbf{x}_n | \mathbf{z}_{1:n-1}). \end{aligned} \quad (11)$$

Let us suppose that each sensor k calculates a (pre-distorted, non-normalized) local pseudoposterior defined as

$$\tilde{f}(\mathbf{x}_n | \mathbf{z}_{1:n-1}, \mathbf{z}_{n,k}) \triangleq f(\mathbf{z}_{n,k} | \mathbf{x}_n) f^{1/K}(\mathbf{x}_n | \mathbf{z}_{1:n-1}). \quad (12)$$

The product of all local pseudoposteriors equals the global posterior up to a factor:

$$\begin{aligned} \prod_{k=1}^K \tilde{f}(\mathbf{x}_n | \mathbf{z}_{1:n-1}, \mathbf{z}_{n,k}) &= \left[\prod_{k=1}^K f(\mathbf{z}_{n,k} | \mathbf{x}_n) \right] f(\mathbf{x}_n | \mathbf{z}_{1:n-1}) \\ &\propto f(\mathbf{x}_n | \mathbf{z}_{1:n}), \end{aligned} \quad (13)$$

where (11) has been used. This posterior reflects all sensor measurements and could be employed as the global PD. However, for a simple distributed computation of (13), we use Gaussian approximations of the local pseudoposteriors and the global posterior, i.e., $\tilde{f}(\mathbf{x}_n | \mathbf{z}_{1:n-1}, \mathbf{z}_{n,k}) \approx \mathcal{N}(\mathbf{x}_n; \tilde{\mu}_{n,k}, \tilde{\mathbf{C}}_{n,k})$ and $f(\mathbf{x}_n | \mathbf{z}_{1:n}) \approx$

¹Note that the global posterior used as PD is different from the global posterior that is obtained by the PF as described in Section 3.

$q(\mathbf{x}_n; \mathbf{z}_n) = \mathcal{N}(\mathbf{x}_n; \mu_n, \mathbf{C}_n)$. Then, using (13) and the rules for a product of Gaussian densities [14], we obtain the following expressions of the mean and covariance of the PD $q(\mathbf{x}_n; \mathbf{z}_n)$:

$$\boldsymbol{\mu}_n = \mathbf{C}_n \sum_{k=1}^K \tilde{\mathbf{C}}_{n,k}^{-1} \tilde{\mu}_{n,k}, \quad \mathbf{C}_n = \left(\sum_{k=1}^K \tilde{\mathbf{C}}_{n,k}^{-1} \right)^{-1}. \quad (14)$$

The sums over all sensors in these expressions can be easily calculated in a distributed way using consensus algorithms.

To calculate the Gaussian approximation $\mathcal{N}(\mathbf{x}_n; \tilde{\mu}_{n,k}, \tilde{\mathbf{C}}_{n,k})$ of the local pseudoposterior $\tilde{f}(\mathbf{x}_n | \mathbf{z}_{1:n-1}, \mathbf{z}_{n,k})$, we note that (12) is the measurement update step of a Bayesian filter using the pre-distorted predicted posterior $f^{1/K}(\mathbf{x}_n | \mathbf{z}_{1:n-1})$ instead of the true predicted posterior $f(\mathbf{x}_n | \mathbf{z}_{1:n-1})$. Furthermore, each sensor calculated a Gaussian approximation of the predicted posterior, $f(\mathbf{x}_n | \mathbf{z}_{1:n-1}) \approx \mathcal{N}(\mathbf{x}_n; \mu'_{n,k}, \mathbf{C}'_{n,k})$ (see Step 2 in Section 3.1); this entails the Gaussian approximation $f^{1/K}(\mathbf{x}_n | \mathbf{z}_{1:n-1}) \approx \mathcal{N}(\mathbf{x}_n; \mu'_{n,k}, K\mathbf{C}'_{n,k})$. Since Gaussian models are thus used for both $\tilde{f}(\mathbf{x}_n | \mathbf{z}_{1:n-1}, \mathbf{z}_{n,k})$ and $f^{1/K}(\mathbf{x}_n | \mathbf{z}_{1:n-1})$, we propose to perform the measurement update in (12) by means of the update step of a *Gaussian filter* [10, 15–17]. This is done locally at each sensor.

The operations of the proposed PD adaptation scheme performed at time n can now be summarized as follows:

Step 1: Each sensor k computes the mean $\tilde{\mu}_{n,k}$ and covariance $\tilde{\mathbf{C}}_{n,k}$ of the Gaussian approximation of the local pseudoposterior, $\mathcal{N}(\mathbf{x}_n; \tilde{\mu}_{n,k}, \tilde{\mathbf{C}}_{n,k}) \approx \tilde{f}(\mathbf{x}_n | \mathbf{z}_{1:n-1}, \mathbf{z}_{n,k})$. This is done locally by performing a Gaussian filter update step with input mean $\mu'_{n,k}$, input covariance $K\mathbf{C}'_{n,k}$, and measurement $\mathbf{z}_{n,k}$. Here, $\mu'_{n,k}$ and $\mathbf{C}'_{n,k}$ were obtained locally according to (5).

Step 2: Consensus algorithms are used to calculate the sums over all sensors in (14). This step requires communication with neighboring sensors. In total, $I[M + M(M+1)/2 + 1]$ real numbers are communicated by each sensor at time n . Here, I denotes the number of consensus iterations and M denotes the dimension of the state \mathbf{x}_n . After convergence of the consensus algorithms, each sensor obtained the global PD $q(\mathbf{x}_n; \mathbf{z}_n) = \mathcal{N}(\mathbf{x}_n; \mu_n, \mathbf{C}_n)$.

5. SIMULATION RESULTS

We consider a target tracking application using acoustic amplitude sensors. The target is represented by the vector $\tau_n = (x_n \ y_n \ \dot{x}_n \ \dot{y}_n)^\top$ containing the target’s 2D position and 2D velocity in the x - y plane. The vector τ_n evolves with time n according to $\tau_n = \mathbf{G}\tau_{n-1} + \mathbf{W}\mathbf{u}'_n$, $n = 1, 2, \dots$, where the matrices $\mathbf{G} \in \mathbb{R}^{4 \times 4}$ and $\mathbf{W} \in \mathbb{R}^{4 \times 2}$ are chosen as in [4] and the \mathbf{u}'_n are independent and identically distributed according to $\mathbf{u}'_n \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_{u'})$ with $\mathbf{C}_{u'} = \text{diag}(0.0033, 0.0033)$. The target motion model specified above is however assumed unknown to the simulated PFs. Therefore, all simulated PFs use a random walk model $\mathbf{x}_n = \mathbf{x}_{n-1} + \mathbf{u}_n$, where the state $\mathbf{x}_n = (x_n \ y_n)^\top$ represents the position of the target and $\mathbf{u}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_u)$ with $\mathbf{C}_u = \text{diag}(0.0528, 0.0528)$ (cf. (1)).

The target emits a sound of constant amplitude $A = 10$, which is sensed by acoustic amplitude sensors. The (scalar) measurement $z_{n,k}$ of sensor k is given by (cf. (2))

$$z_{n,k} = \frac{A}{\|\mathbf{x}_n - \xi_{n,k}\|^2} + v_{n,k},$$

where $\xi_{n,k}$ is the position of sensor k at time n and $v_{n,k} \sim \mathcal{N}(0, \sigma_v^2)$ with $\sigma_v^2 = 0.00005$. This value of σ_v^2 yields a peaky likelihood, which highlights the performance gains of PD adaptation. The network consists of $K = 25$ sensors that are deployed on a jittered grid within a rectangular region of size $40\text{m} \times 40\text{m}$. Each sensor communicates with other sensors within a range of 18m .

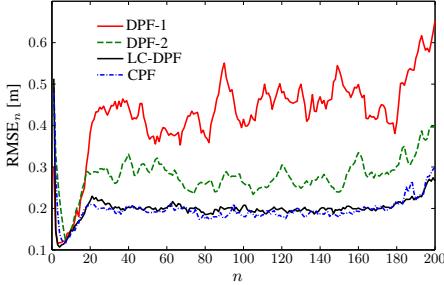


Fig. 1. RMSE_n versus time n .

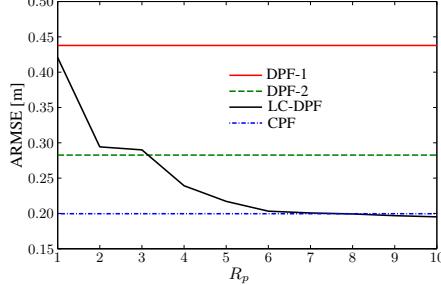


Fig. 2. ARMSE versus the degree of polynomial approximation R_p .

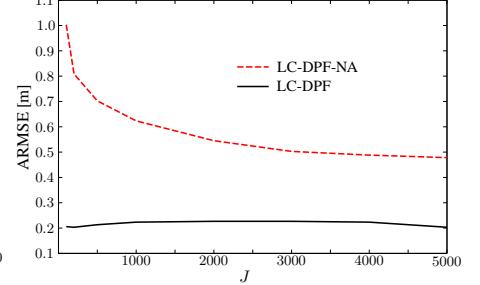


Fig. 3. ARMSE versus the number of particles J .

For LC, unless stated otherwise, we approximate $\log f(\mathbf{z}_{n,k} | \mathbf{x}_n)$ by a multivariate polynomial of degree $R_p = 6$; this leads to a basis expansion (7) of degree $R = \binom{R_p+M}{R_p} = 28$. The sums in (9) and (14) are computed by $I = 15$ iterations of an average consensus algorithm with Metropolis weights [18]. For PD adaptation (Step 1 in Section 4), the update step of an unscented Kalman filter [15] is used.

We compare the proposed LC-DPF with the distributed PFs presented in [7] and [8] (referred to as DPF-1 and DPF-2, respectively) and with a centralized PF (CPF) that processes all sensor measurements at a fusion center. The CPF uses an adapted PD that is computed using a (centralized) unscented Kalman filter. The number of particles at each sensor of the distributed PFs and at the fusion center of the CPF is $J = 200$ unless stated otherwise. As a performance measure, we use the root-mean-square error of the state estimates $\hat{\mathbf{x}}_{n,k}$, denoted RMSE_n , which is computed as the square root of the average of the squared estimation error over all sensors and over 1000 simulation runs. We also compute the *average* RMSE (ARMSE) by averaging RMSE_n^2 over all 200 simulated time instants n and taking the square root of the result.

Fig. 1 shows the temporal evolution of RMSE_n . It can be seen that the performance of LC-DPF is almost as good as that of CPF and better than that of DPF-1 and DPF-2. The communication requirements of LC-DPF are lower than those of DPF-1 but higher than those of DPF-2: the total counts of real numbers transmitted by LC-DPF, DPF-1, and DPF-2 during one time step in the entire network (all sensors) are 12375, 76875, and 1875, respectively.

Fig. 2 shows the ARMSE versus the degree R_p of the polynomial used to approximate the local log-likelihood functions. As expected, the ARMSE of LC-DPF decreases with growing R_p and approaches that of CPF. Note, however, that the communication requirements increase with growing R_p .

Finally, Fig. 3 shows the dependence of the ARMSE on the number J of particles for LC-DPF and an LC-DPF without proposal adaptation (abbreviated as LC-DPF-NA) [4]. As we can see, the ARMSE of LC-DPF is significantly lower than that of LC-DPF-NA, even if J is large. This demonstrates the performance improvement achieved by our distributed PD adaptation scheme.

6. CONCLUSION

We presented a consensus-based distributed particle filter (PF) for wireless sensor networks. The state estimates computed by the local PFs at the various sensors reflect the past and present measurements of all sensors. This is enabled by a generalized likelihood consensus scheme, which performs a distributed approximate calculation of the joint likelihood function for general measurement models. Our main contribution was a distributed method for adapting the proposal density (PD) used by the local PFs. This method is based on a Gaussian model for the PD, whose mean and covariance are computed in a distributed way by consensus algorithms and by the update step of a Gaussian filter. Simulation results demonstrated the good per-

formance of the proposed distributed PF and the large performance gains achieved by the proposed PD adaptation method.

7. REFERENCES

- [1] A. Doucet, N. De Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*. New York, NY: Springer, 2001.
- [2] O. Hlinka, O. Slučiak, F. Hlawatsch, P. M. Djurić, and M. Rupp, “Likelihood consensus: Principles and application to distributed particle filtering,” in *Proc. 44th Asilomar Conf. Sig., Syst., Comp.*, (Pacific Grove, CA), pp. 349–353, Nov. 2010.
- [3] O. Hlinka, O. Slučiak, F. Hlawatsch, P. M. Djurić, and M. Rupp, “Distributed Gaussian particle filtering using likelihood consensus,” in *Proc. IEEE ICASSP-11*, (Prague, Czech Rep.), pp. 3756–3759, May 2011.
- [4] O. Hlinka, O. Slučiak, F. Hlawatsch, P. M. Djurić, and M. Rupp, “Likelihood consensus and its application to distributed particle filtering.” Available online: arXiv:1108.6214v1 [stat.AP], Aug. 2011.
- [5] M. Šimandl and O. Straka, “Sampling densities of particle filter: A survey and comparison,” in *Proc. IEEE ACC-07*, (New York, NY), pp. 4437–4442, Jul. 2007.
- [6] R. van der Merwe and E. Wan, “Gaussian mixture sigma-point particle filters for sequential probabilistic inference in dynamic state-space models,” in *Proc. ICASSP-03*, (Hong Kong, China), pp. 701–704, Apr. 2003.
- [7] S. Farahmand, S. I. Roumeliotis, and G. B. Giannakis, “Set-membership constrained particle filter: Distributed adaptation for sensor networks,” *IEEE Trans. Signal Process.*, vol. 59, pp. 4122 – 4138, Sep. 2011.
- [8] A. Mohammadi and A. Asif, “Consensus-based distributed unscented particle filter,” in *Proc. IEEE SSP-11*, (Nice, France), pp. 237–240, Jun. 2011.
- [9] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Upper Saddle River, NJ: Prentice-Hall, 1993.
- [10] H. Tanizaki, *Nonlinear Filters: Estimation and Applications*. Berlin, Germany: Springer, 1996.
- [11] R. Olfati-Saber, J. A. Fax, and R. M. Murray, “Consensus and cooperation in networked multi-agent systems,” *Proc. IEEE*, vol. 95, pp. 215–233, Jan. 2007.
- [12] Å. Björck, *Numerical Methods for Least Squares Problems*. Philadelphia, PA: SIAM, 1996.
- [13] B. N. Oreshkin and M. J. Coates, “Asynchronous distributed particle filter via decentralized evaluation of Gaussian products,” in *Proc. ISIF Int. Conf. Inform. Fusion*, (Edinburgh, UK), pp. 1–8, Jul. 2010.
- [14] M. J. F. Gales and S. S. Airey, “Product of Gaussians for speech recognition,” *Computer Speech & Language*, vol. 20, pp. 22–40, Jan. 2006.
- [15] R. van der Merwe, *Sigma-point Kalman filters for probabilistic inference in dynamic state-space models*. PhD thesis, Oregon Health and Science University, Apr. 2004.
- [16] I. Arasaratnam and S. Haykin, “Cubature Kalman filters,” *IEEE Trans. Autom. Contr.*, vol. 54, pp. 1254–1269, Jun. 2009.
- [17] K. Ito and K. Xiong, “Gaussian filters for nonlinear filtering problems,” *IEEE Trans. Autom. Contr.*, vol. 45, pp. 910–927, May 2000.
- [18] L. Xiao, S. Boyd, and S. Lall, “A scheme for robust distributed sensor fusion based on average consensus,” in *Proc. IEEE IPSN-05*, (Los Angeles, CA), pp. 63–70, Apr. 2005.